

Figure 1. System Configuration of Clustered Video Server System

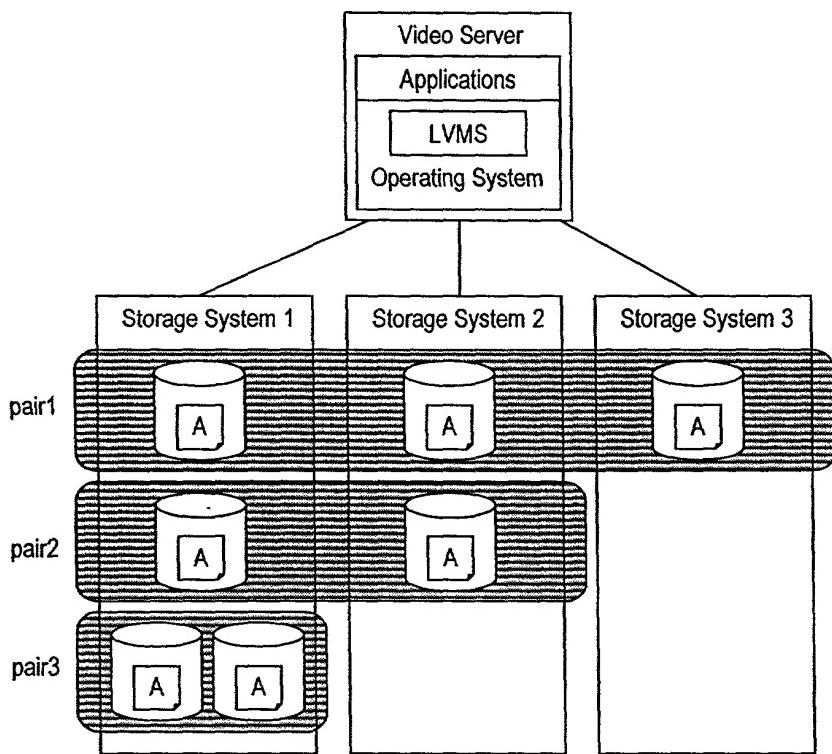


Figure 2. Example of Disk Mirroring on Multiple Storage Systems by LVMS

16A 16B 16C 10 12

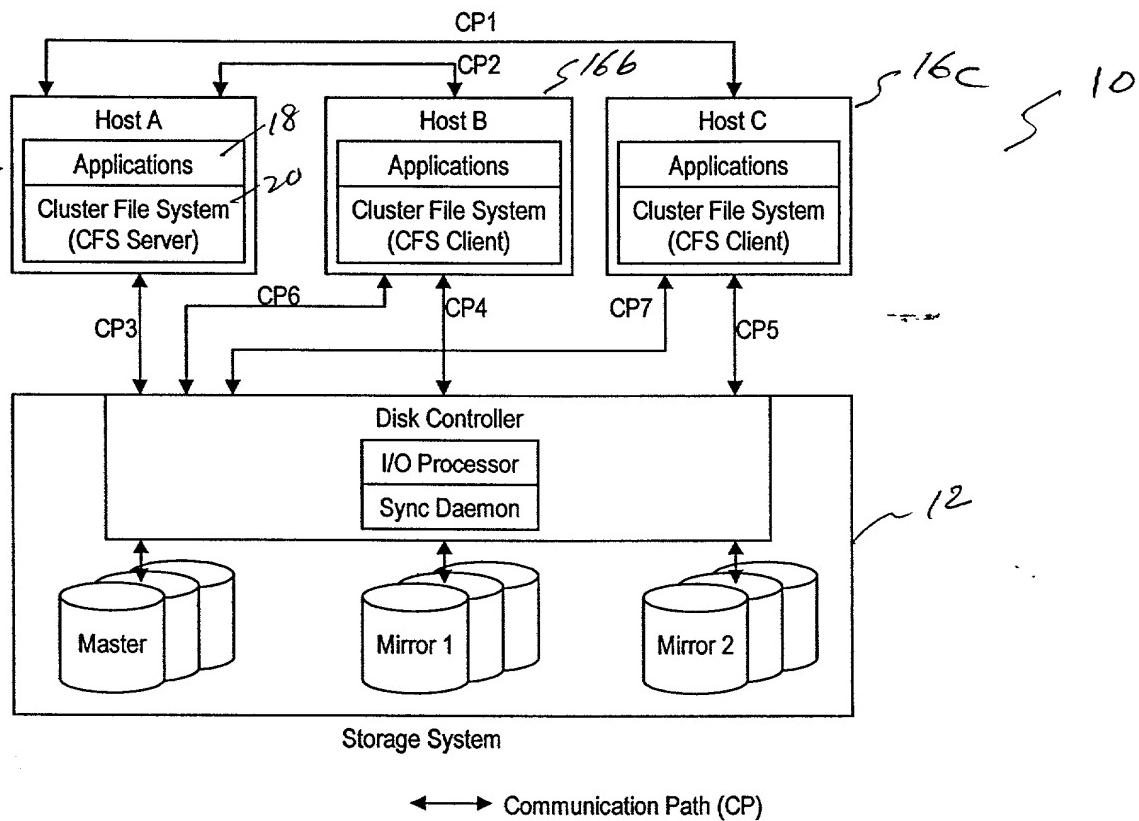


Figure 3 System Configuration where Mirrors in the Same Storage System

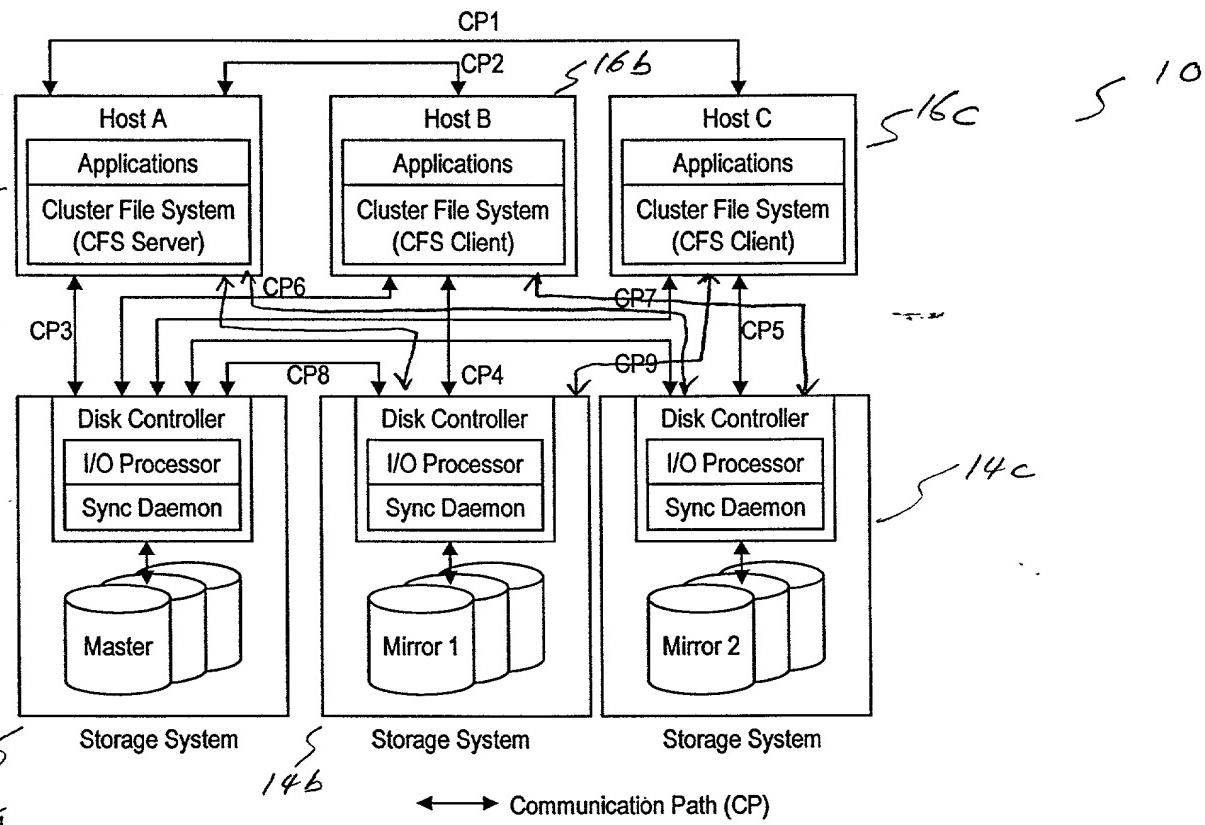


Figure 4. System Configuration of Second Description

Pair Name	Master		Mirror 1		Mirror 2	
	SS ID	Vol ID	SS ID	Vol ID	SS ID	Vol ID
pair1	1	8	2	2	3	5
pair2	1	12	1	7	N/A	N/A
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Figure 5. Pair Configuration Table

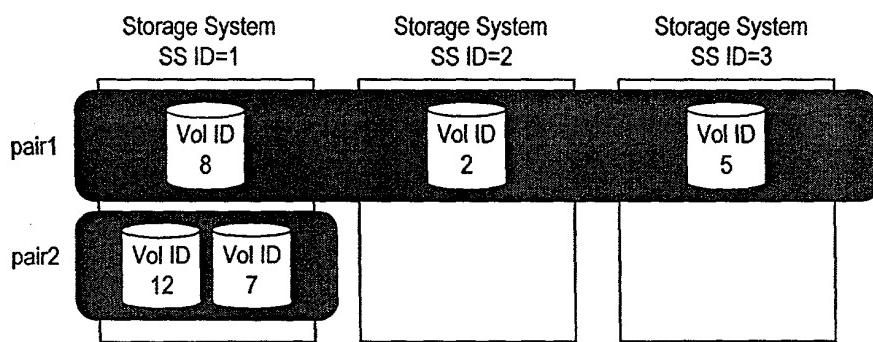


Figure 6. Graphical Understanding about Mirroring Configuration

Pair Name	Master	Mirror 1	Mirror 2
pair1	100	200	50
pair2	300	10	N/A
⋮	⋮	⋮	⋮

Figure 9. Usage Table of Pairs

File Name	Block #1			Block #2			Block #3			.....
	SS ID	Vol ID	Offset	SS ID	Vol ID	Offset	SS ID	Vol ID	Offset	
file1	1	8	100	1	8	700	1	8	900	.....
file2	1	8	200	1	8	150	1	8	600	.....
file3	1	12	10	1	12	80	1	12	100	.....
...	...	...	...	...	...	...	...	...	...	.....

Figure 7. File Allocation Table of CFS Server

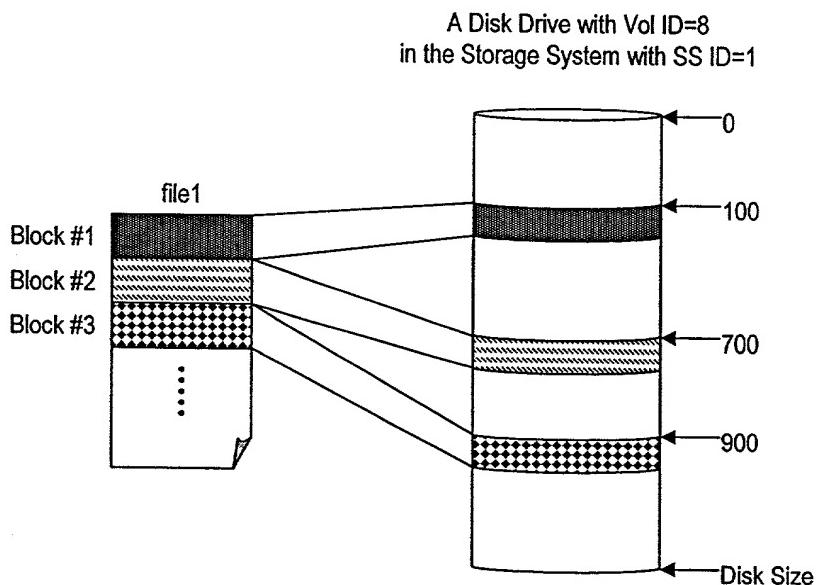


Figure 8. Graphical Understanding about File Allocation List

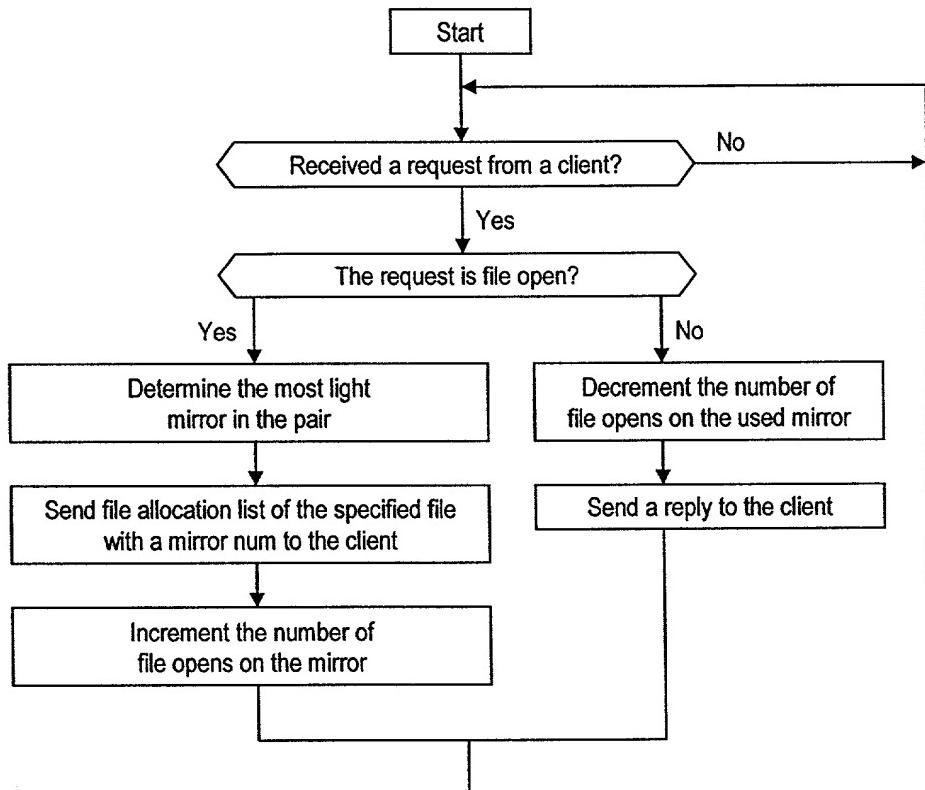


Figure 10 Process Sequence of CFS Server

```
struct {
    int mirror_num;
    int block_num;
    struct {
        int master_ss_id;
        int master_vol_id;
        int offset;
    } block_list[block_num];
} file_allocation_list
```

Figure 11. The Format of File Allocation List

```
struct {
    int type;
    union {
        struct {
            char *filename;
            int mode;
        } open;
        struct {
            int file_id;
            int offset;
            int size;
            char *buf;
        } write;
        struct {
            int file_id;
            int offset;
            int size;
            char *buf;
        } read;
        struct {
            int file_id;
        } close;
    } u;
} file_io_request
```

Figure 12 .File I/O Request Format

#define	File_Open	1
#define	File_Write	2
#define	File_Read	3
#define	File_Close	4

Figure 13 .File I/O Type

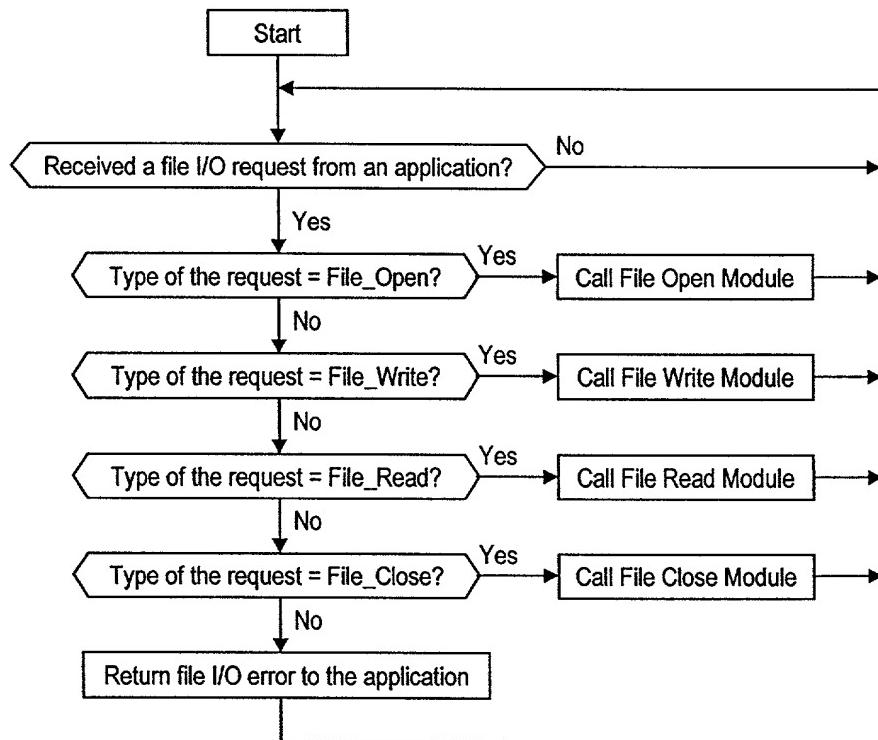


Figure 14 Process Sequence of CFS Client

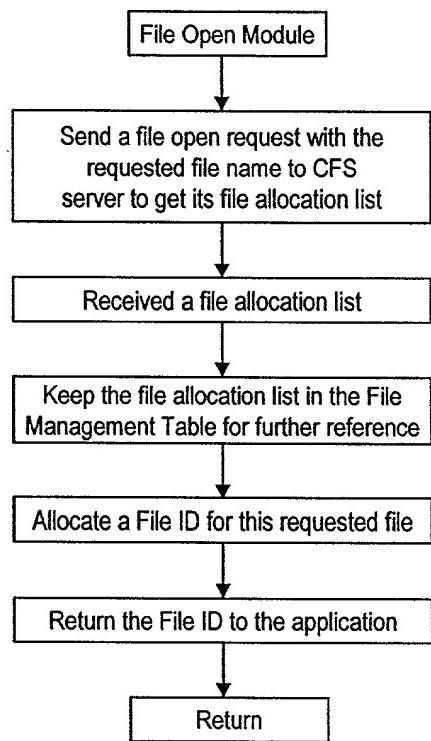


Figure 15. File Open Module

```
struct {  
    struct file_allocation_list *fal;  
} file_mgmt_table[max_file_id];
```

Figure 16. File Management Table

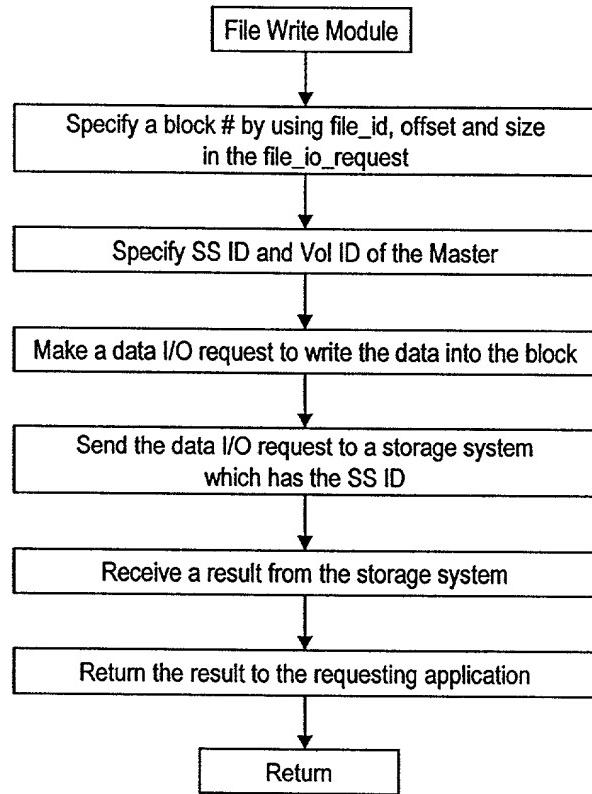


Figure 17. File Write Module

```
struct {  
    int type;  
    int vol_id;  
    int offset;  
    int size;  
    char *data;  
} data_io_request
```

Figure 18. Data I/O Request

```
#define Data_Read      1  
#define Data_Write     2
```

Figure 19. Types of Data I/O Request

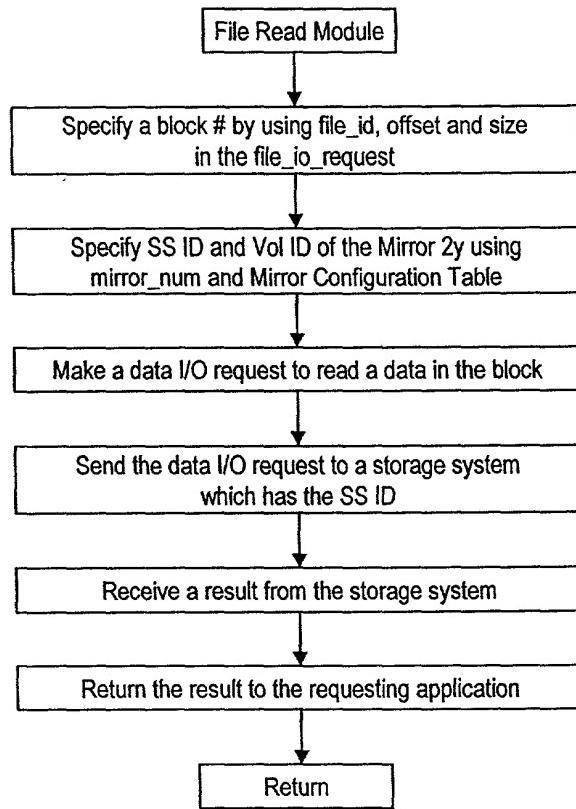


Figure 20 File Read Module

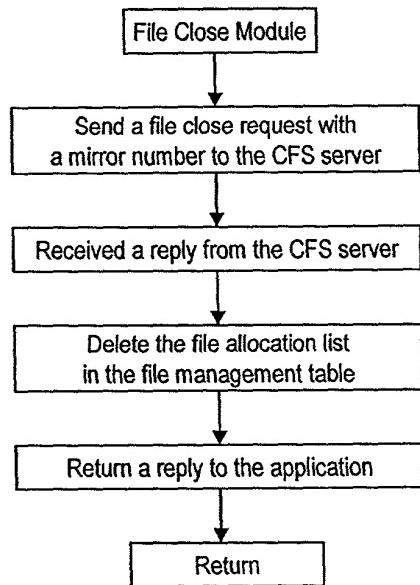


Figure 2.1 File Close Module

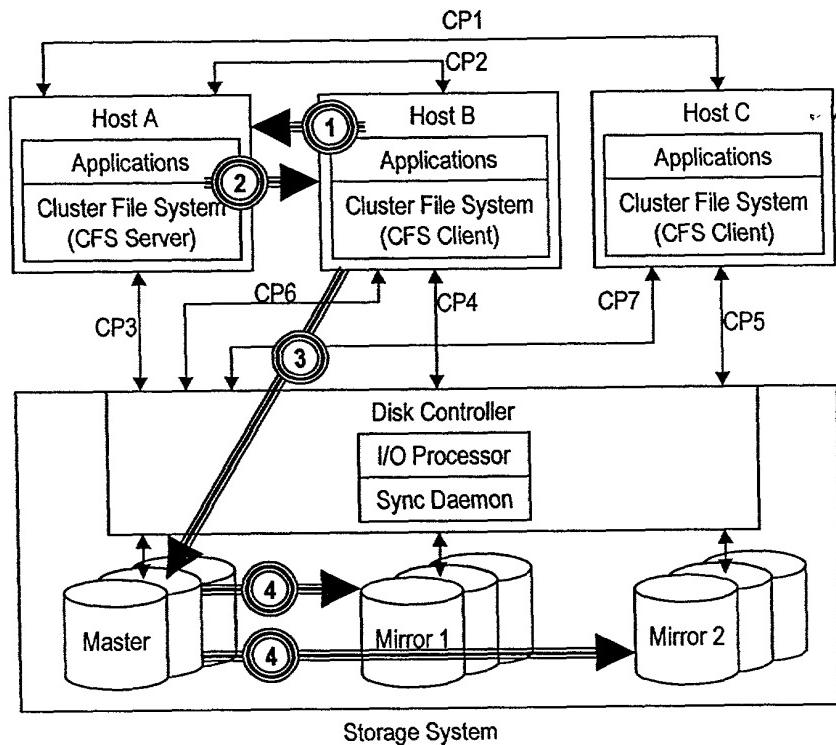


Figure 30 Synchronous Data Write Sequence where Mirrors in the Same Storage System

Data Write Sequence

- ① Request File Allocation List
- ② Return File Allocation List
- ③ Write Data to Master
- ④ Copy Data to Mirrors by Sync Daemon

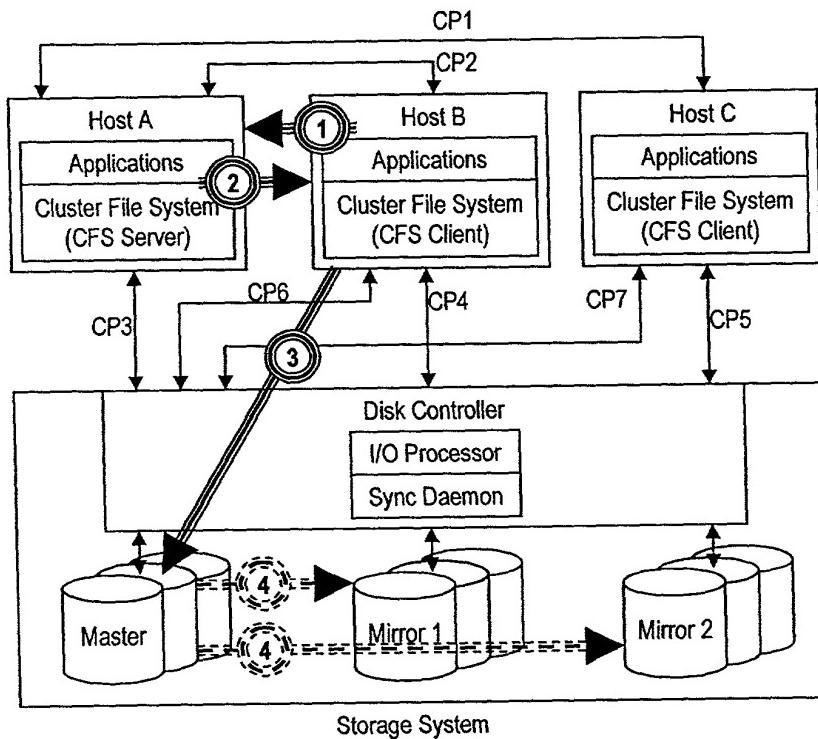


Figure 23 Asynchronous Data Write Sequence where Mirrors in the Same Storage System

Data Write Sequence

- 1 Request File Allocation List
- 2 Return File Allocation List
- 3 Write Data to Master
- 4 Copy Data to Mirrors by Sync Daemon

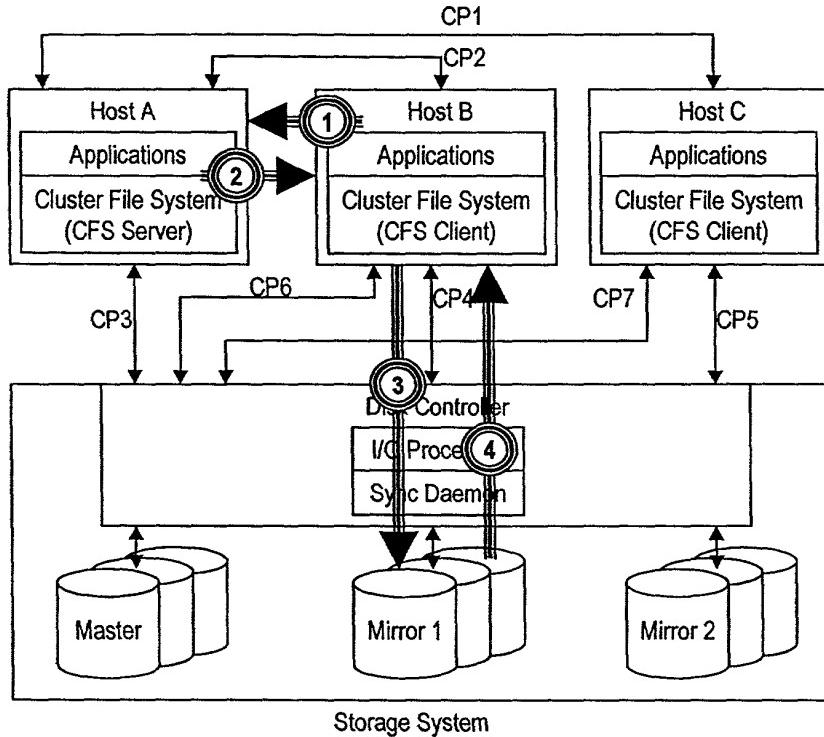


Figure 24 Data Read Sequence where Mirrors in the Same Storage System  
(A Case of Synchronous Data Write or Async Data Write and Data is on Mirror)

Data Read Sequence

- ① Request File Allocation List
- ② Return File Allocation List
- ③ Read Data from Mirror 1
- ④ Send Data to Host

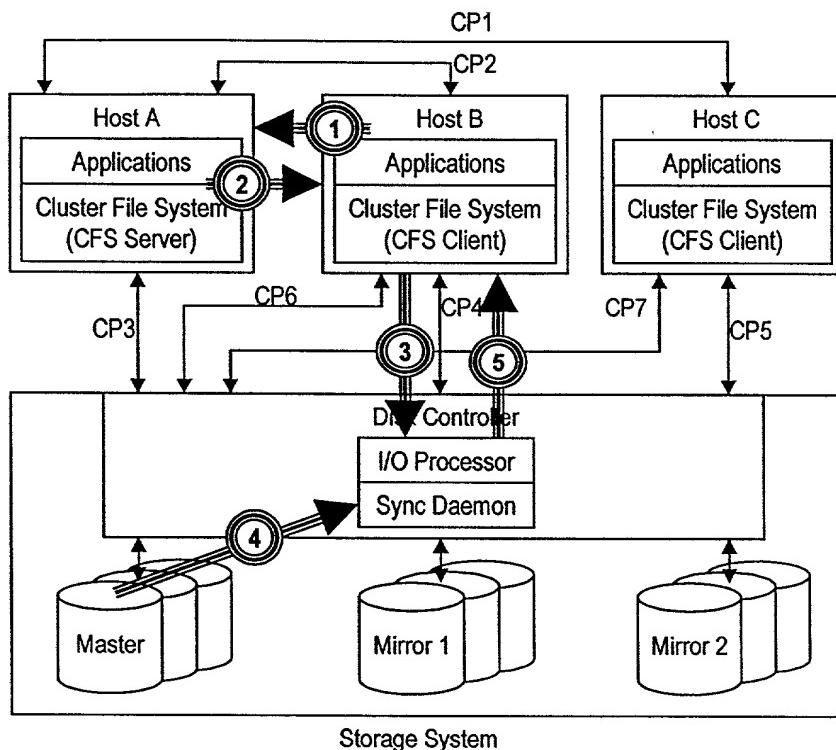


Figure 25 Data Read Sequence where Mirrors in the Same Storage System  
(A Case of Asynchronous Data Write and Data is not on Mirror)

Data Read Sequence

- ① Request File Allocation List
- ② Return File Allocation List
- ③ Read Data from Mirror 1
- ④ Read Data from Master by I/O Processor
- ⑤ Send Data to Host

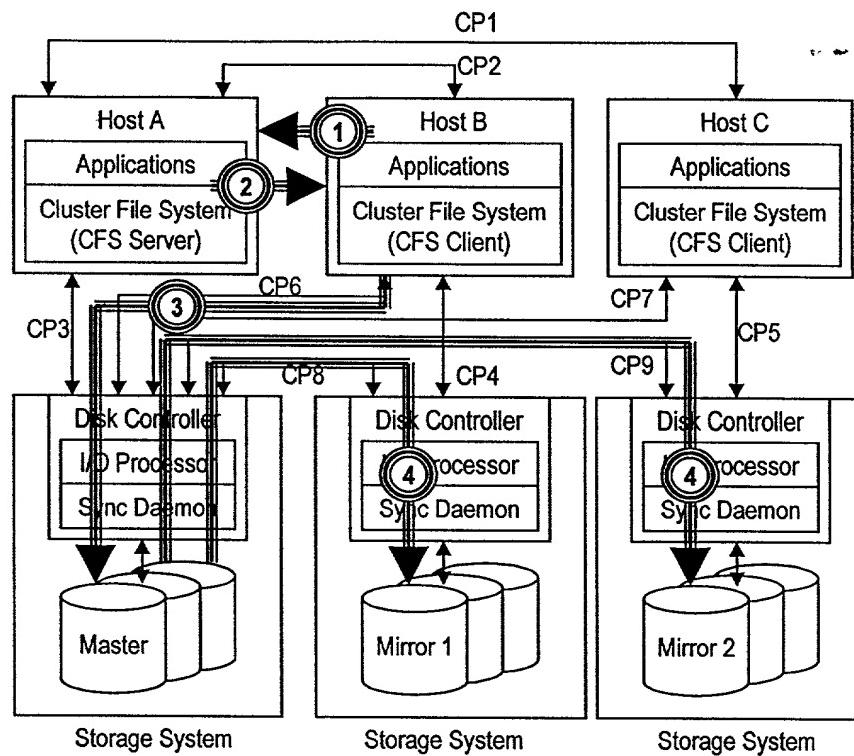


Figure 2-6 Synchronous Data Write Sequence where Mirrors on Different Storage Systems

Data Write Sequence

- ① Request File Allocation List
- ② Return File Allocation List
- ③ Write Data to Master
- ④ Copy Data to Mirrors through Communication Path

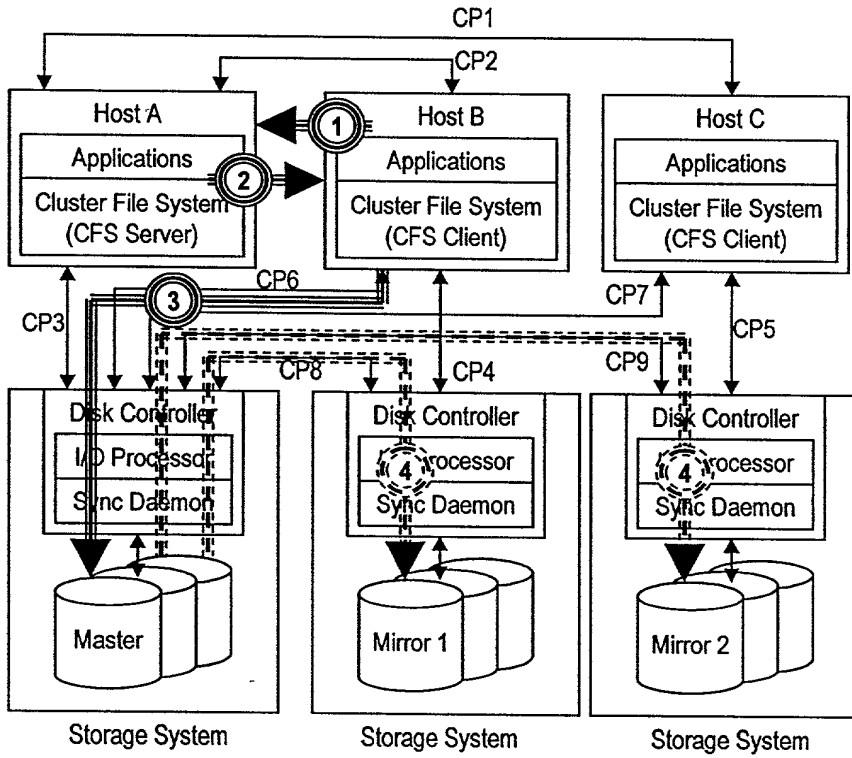


Figure 27 Asynchronous Data Write Sequence where Mirrors on Different Storage Systems

Data Write Sequence

- ① Request File Allocation List
- ② Return File Allocation List
- ③ Write Data to Master
- ④ Copy Data to Mirrors through Communication Path by Sync Daemon

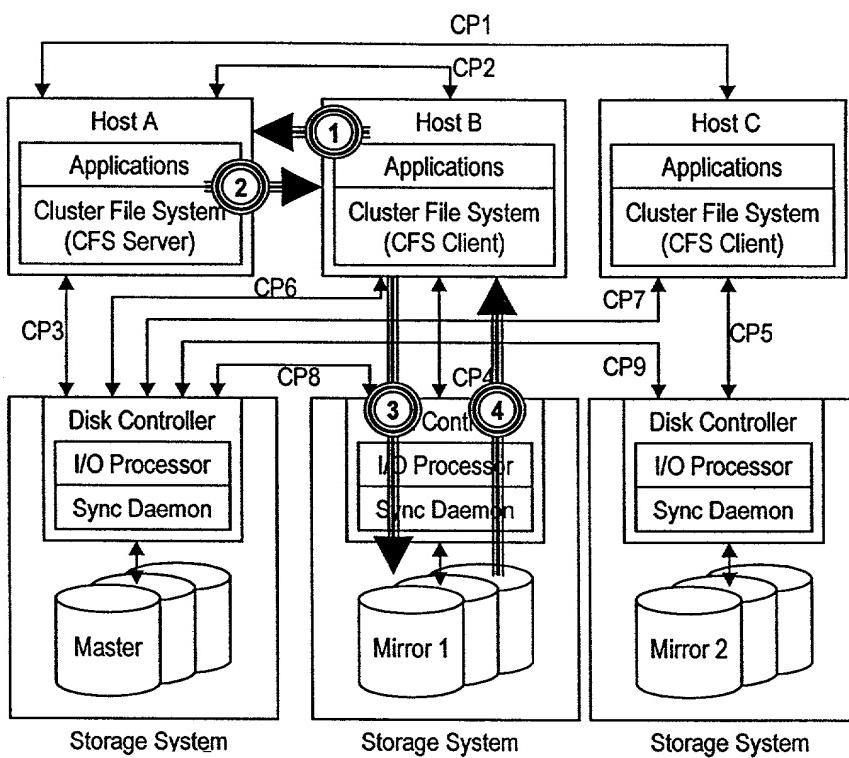


Figure 28 Data Read Sequence where Mirrors on Different Storage Systems  
(A Case of the Synchronous Data Write or Async Data Write and Data is on Mirror)

Data Read Sequence

- 1 Request File Allocation List
- 2 Return File Allocation List
- 3 Read Data from Mirror 1
- 4 Send Data to Host

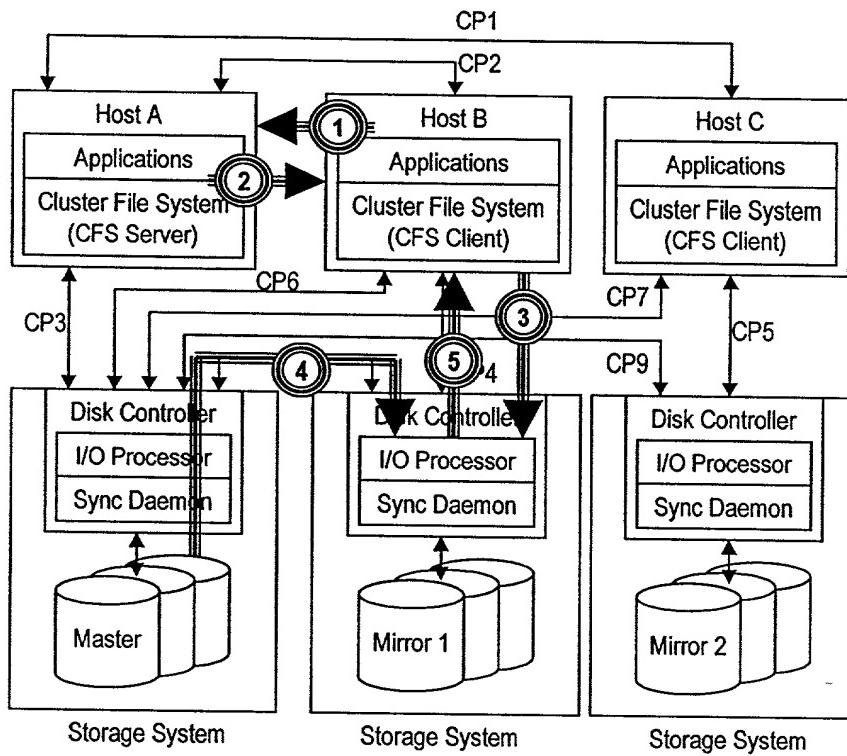


Figure 2.9 Data Read Sequence where Mirrors on Different Storage Systems  
(A Case of the Asynchronous Data Write and Data is not on Mirror)

Data Read Sequence

- ① Request File Allocation List
- ② Return File Allocation List
- ③ Read Data from Mirror 1
- ④ Read Data from Master by Sync Client
- ⑤ Send Data to Host

		Same Storage System	Different Storage Systems
Sync Data Write	Data Write	Figure 22	Figure 26
	Data Read	Figure 24	Figure 28
Async Data Write	Data Write	Figure 23	Figure 27
	Data Read	Data is in Mirror	Figure 28
		Data is not in Mirror	Figure 29

Figure 30 Relationship between Figures and Read/Write Cases

Pair Name	Mirror ID	SS ID	Vol ID	Cluster #1	Cluster #2	Cluster #3	.....
pair1	Mirror 1	1	1	Valid	Invalid	Valid	.....
	Mirror 2	2	2	Invalid	Invalid	Valid	.....
	...	...	...	...	...	...	.....
	Mirror N	2	2	Invalid	Valid	Valid	.....
pair2	Mirror 1	1	1	Valid	Valid	Valid	.....
...	...	...	...	...	...	...	.....

Figure 31. Bitmap Table

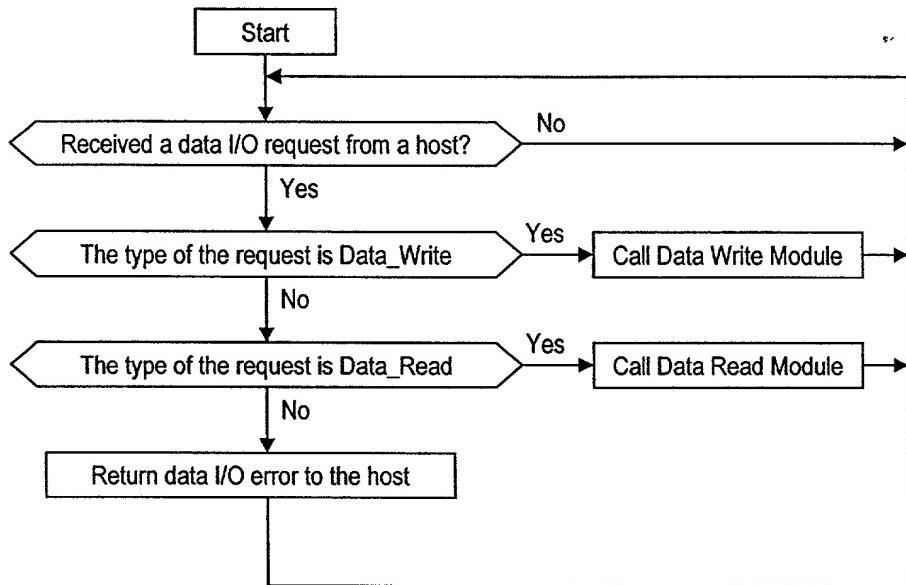


Figure 32. Process Sequence of I/O Processor

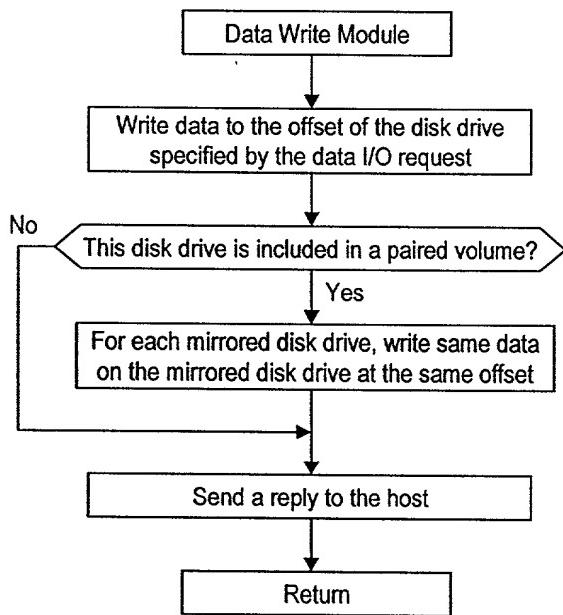


Figure 33. Data Write Module for Synchronous Data Write

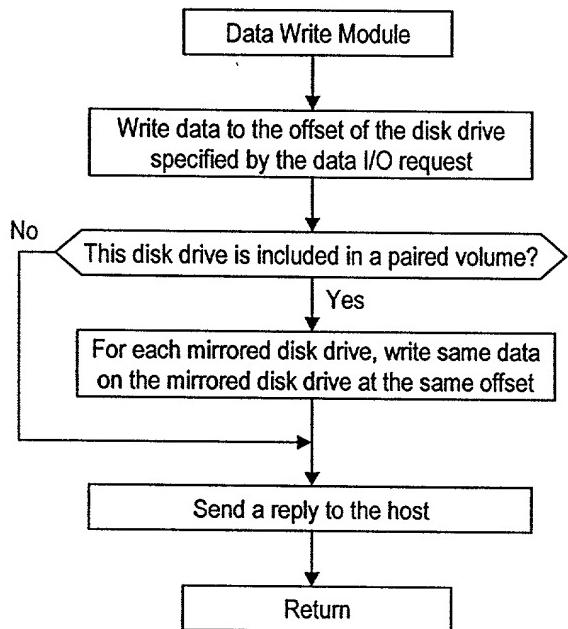


Figure 33. Data Write Module for Synchronous Data Write

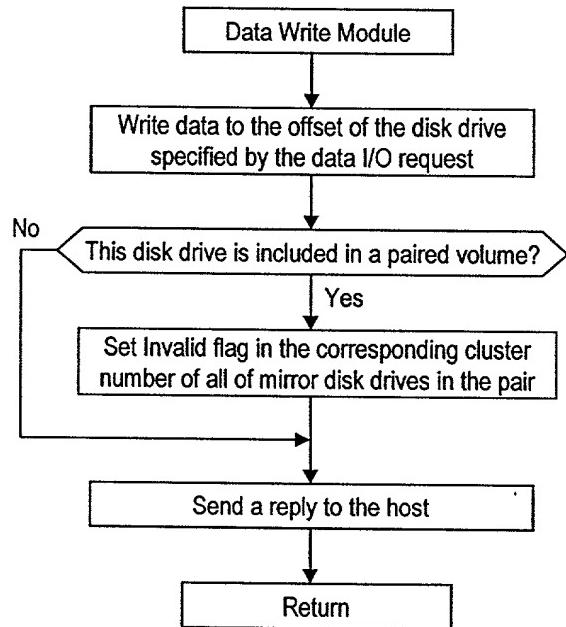


Figure 34. Data Write Module for Asynchronous Data Write

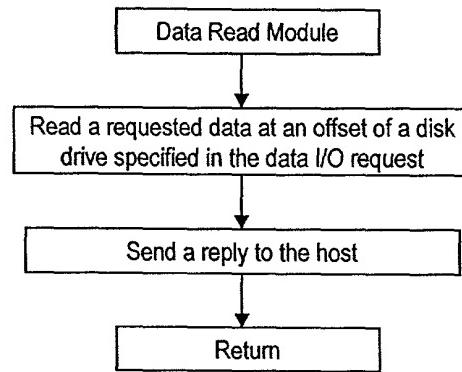


Figure 35. Data Read Module for Synchronous Write

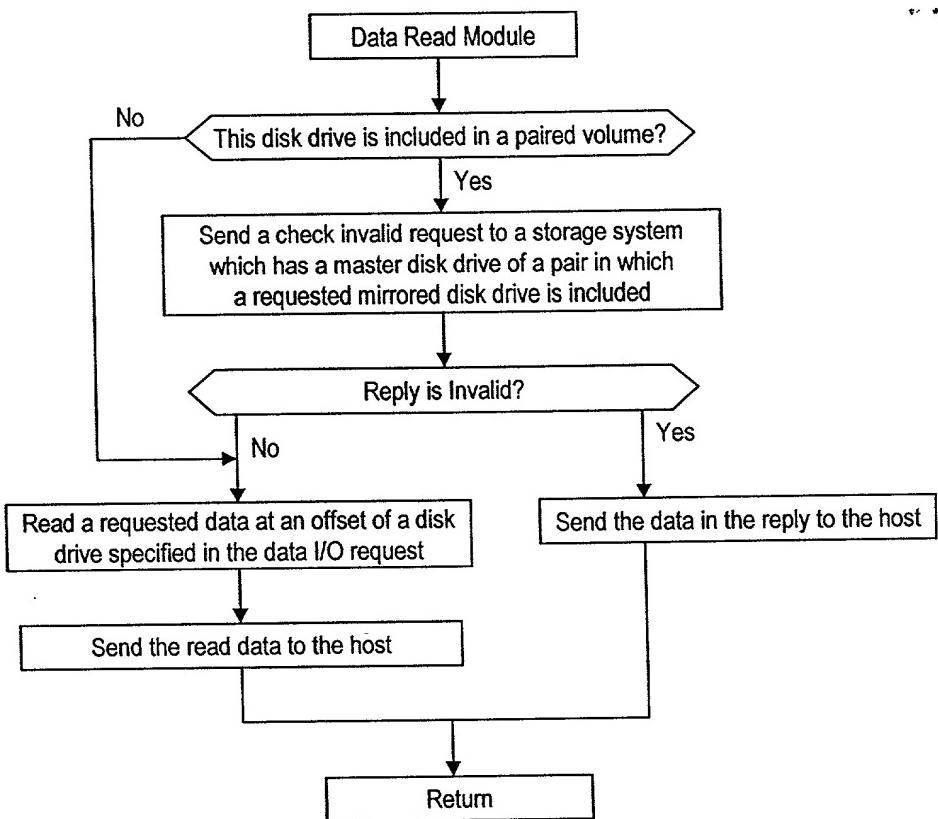


Figure 36. Data Read Module for Asynchronous Write

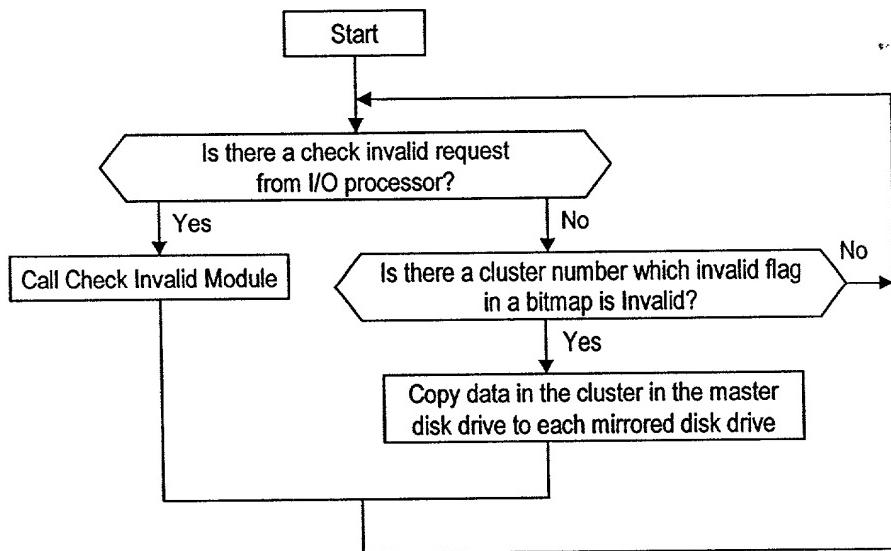


Figure 37. Process Sequence of Sync Daemon

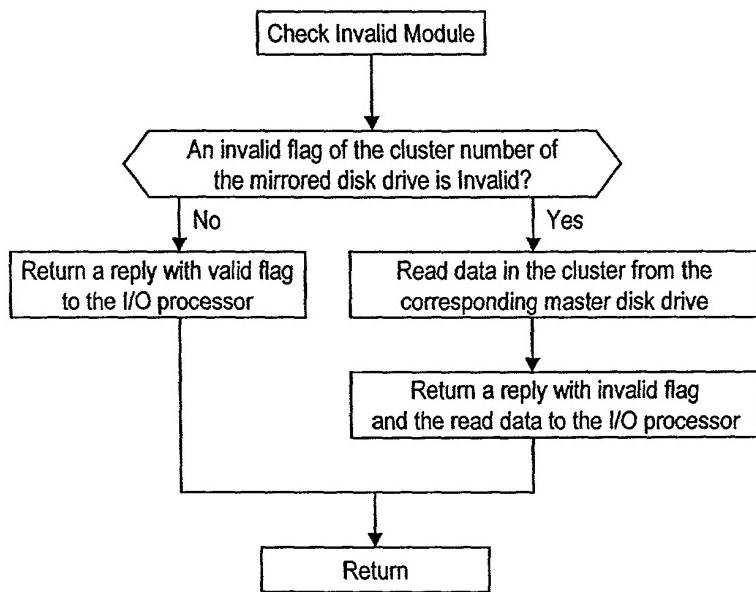


Figure 38. Check Invalid Module

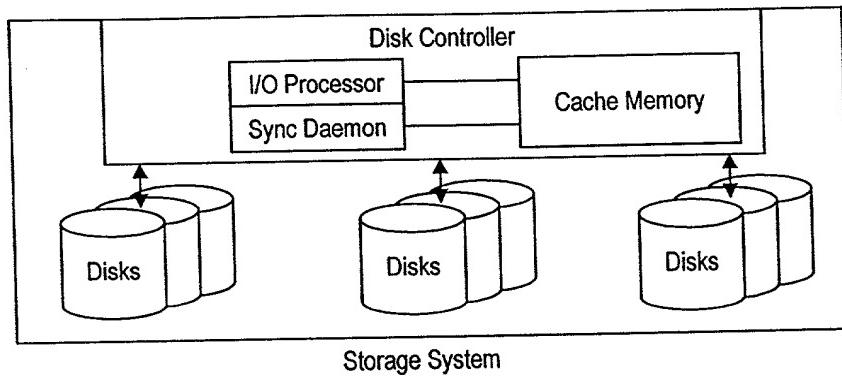


Figure 39. Storage System with Cache Memory